

Interaktion

April 1985

IUGN-7

CONTENTS

	Page
EDITORIAL	2
COUNTMEN	3
IQT-1	5
CP/M ON THE INTERAK (PART 2)	6
SOFTWARE COMPETITION INFORMATION	16
BOOK LIBRARY LISTING	16
SOFTWARE LIBRARY LISTINGS	18
GENERAL PURPOSE VDU2K DRIVER	19
LETTERS	21

CHAIRMAN	Mr P.P.Vella.
SOFTWARE LIBRARIAN	Mr P.P.Vella.
MEMBERSHIP SECRETARY	Mr P.P.Vella.
IUGN EDITOR	Mr R.Eldridge.
MAILING LISTS	Mr R.Eldridge.
BOOK LIBRARIAN	Mr R.E.Bowyer.
PRINTING	Mr D.M.Parkins.
DISTRIBUTION	Mr D.M.Parkins.

IUG is the title of the INTERAK USER GROUP.
IUGN is the newsletter of the IUG.

Editorial

Dear Member,

Spring is with us again, presumably you are contemplating the return to another years gardening and decorating. Whilst you are out there please spare a thought for the magazine and consider writing a contribution for a future issue. We lack stories about how you finally got it to work and what you are doing with it. If you are inspired you can now send your work directly to me at :-

28 Wycherley Close,
Blackheath,
London,
SE3 7QH.

Please don't telephone me as I am on call from home for my work, and if my phone is hung up I can't work, pay bills or prepare newsletters.

A backbench bill sponsored by Mr William Powel (Corby, C) recieved an unopposed second reading in the house on the 22 February 1985. This bill, The Copyright Computer Software Amendment Bill, is designed to prevent computer software piracy by applying the copyright law to computer programs, including those made before the bill was introduced. The bill will give the same protection to software as is applied to literary works under the Copyright Act 1956. Penalties will be a maximum sentence of two years imprisonment and unlimited fines.

David tells me that a tracked version of the VDU-2K is in the early stages of production. This will have a no-snow circuit to allow the same flexibility of programming that the VDU-K does.

A member recently told me a story about how he and another member went to a computer club run by a university graduate, whose wise words to him (regarding the clubs hard disk) were :-

"Don't take any notice of it, it's a magic piece of equipment, and you will never understand it; forget that it is there."

What a wally! I don't think the member will bother to go again, and will probably try to forget that he was there.

Mr R.J.Swaine has written to David at Greenbank about the CUSTOM 80 system, wondering if our newsletter could be used as a medium by other CUSTOM 80 users. We see no objection to this and therefore invite contributions from CUSTOM 80 users. If there is sufficient interest I will try to set up a CUSTOM 80 section especially for them. Of interest to such users is the rumour that some INTERAK users have CUSTOM 80 cards modified to work on their systems. One has even altered ZYMON to drive the 40 character CUSTOM 80 VDU card.

COUNTMEN

by It's a Cop Out Dave Gordon

For ZYBASIC 2 with a 32 column 24 line screen

A counting game for children, with a picture as reward

(ED- Our printer prints £ for the hash symbol)

```
20 RESTORE
30 E1=0
70 CLS
80 ?"HELLO WHAT IS YOUR NAME"
90 A$=INPUT$
100 D1=0: A1=0
110 C1=£1D
120 B1=RND(9)
130 IF B1=0 GOTO 120
135 PAGE: LINE1
140 GOSUB 1000
150 ?"HOW MANY LITTLE MEN CAN YOU SEE"
160 ?A$
170 INK.A1
175 A1=A1-48
180 IF A1=-48 GOTO 170
190 IF A1=B1 GOSUB 2000
200 IF A1<>B1 GOSUB 3000
241 IF E1>7 GOTO 20
243 IF D1=3 GOTO 250
245 IF A1<>B1 GOTO 140
250 GOTO 100
999 STOP
1000 CLS
1010 FOR N=1 TO B1*3 STEP 3: POKE £F182+N,C1: NEXT N
1999 RETURN
2000 CLS: PAGE: LINES
2005 RESTORE
2010 E1=E1+2
2020 READ Z$
2030 ?Z$
2040 READ X$
2050 LINE 12: ?X$
2093 FOR F=1 TO 500: NEXT F
2095 POKE £F19C-E1*33,£20
2097 FOR F=1 TO 500: NEXT F
2210 POKE £F19C-E1*33,C1
2400 LINE 21:
2401 ?"PRESS ANY KEY TO CONTINUE"
2500 IF E1>8 GOSUB 8000
2905 INK.R1
2907 IF E1<8 GOTO 2920
2910 IF R1=0 GOSUB 8000
2920 IF R1=0 GOTO 2093
2999 RETURN
3000 CLS
3010 D1=D1+1
3015 FOR N1=1 TO 4
3020 FOR N=1 TO 330 STEP 33: POKE £F10B+N,£0F: NEXT N
3050 FOR N=1 TO 310 STEP 31: POKE £F114+N,£0F: NEXT N
```

```

3075 FOR F=1 TO 500: NEXT F
3080 CLS
3090 NEXT N1
3100 IF D1<3 GOTO 3999
3110 PAGE
3120 ?"HARD LUCK THE KEY TO PRESS WAS"
3130 FOR N=1 TO 80: ?B1,: NEXT N
3160 ?"PRESS ANY KEY TO CONTINUE"
3170 INK.Z6
3180 IF Z6=0 GOTO 3170
3999 RETURN
8000 FOR N3=1 TO 10
8010 POKE £F095-N3,C1
8020 FOR F=1 TO 40: NEXT F
8035 POKE £F095-N3,£20
8040 NEXT N3
8050 FOR N3=1 TO 6
8053 POKE £F08A+N3*32,C1
8055 FOR F=1 TO 40: NEXT F
8060 POKE £F08A+N3*32,£20
8080 NEXT N3
8090 FOR N3=1 TO 5
8100 POKE £F14A-N3*32,£2A
8110 FOR F=1 TO 40: NEXT F
8125 POKE £F14A-N3*32,£20
8126 POKE £F14A,£7E
8130 NEXT N3
8999 RETURN
9100 DATA" please use the table below for how to type this line"
9110 DATA" use the table again and put quotes at end of each line"

```

-----LINE 9100-----	TABLE	-----LINE 9110-----
Line 9100 is as follows:-		Line 9110 is as follows:-
45 spaces (ASCII 20H)		1 control-0 (ASCII 0FH)
6 underlines (ASCII 5FH)		18 spaces (ASCII 20H)
2 control-0 (ASCII 0FH)		14 control-0 (ASCII 0FH)
30 spaces (ASCII 20H)		18 spaces (ASCII 20H)
3 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
29 spaces (ASCII 20H)		12 spaces (ASCII 20H)
4 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
9 spaces (ASCII 20H)		18 spaces (ASCII 20H)
1 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
18 spaces (ASCII 20H)		12 spaces (ASCII 20H)
5 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
8 spaces (ASCII 20H)		18 spaces (ASCII 20H)
1 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
18 spaces (ASCII 20H)		12 spaces (ASCII 20H)
6 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
7 spaces (ASCII 20H)		18 spaces (ASCII 20H)
1 control-0 (ASCII 0FH)		1 control-0 (ASCII 0FH)
18 spaces (ASCII 20H)		12 spaces (ASCII 20H)
7 control-0 (ASCII 0FH)		20 control-0 (ASCII 0FH)

THE END

IQT-1

A TEST OF INTELLIGENCE AND OBSERVATION

You are allowed five minutes to complete this paper.

Work your answers in the space provided preceeding each item number.

Work as quickly as you can.

A man of average intelligence will complete the paper in 4 minutes.

- 1) I went to bed at eight o'clock in the evening and set the alarm to get up at nine in the morning, how many hours sleep will this allow me?
- 2) Do they have a 4th July in England?
- 3) Some months have 30 days, some have 31. How many have 28 days?
- 4) If you had only one match and entered a dark room where there was an oil lamp, oil heater and some kindleing wood. Which would you light first?
- 5) If a doctor gave you three pills and told you to take one every half hour, how long would they last?
- 6) A farmer has 17 sheep. All but 9 die, how many did he have left?
- 7) Divide 30 by $1/2$. Add 10. What is the answer?
- 8) A man builds a house with four sides, a rectangular structure each having a southern exposure. A big bear came wandering by. What colour is the bear?
- 9) How many animals of each species did Moses take aboard the Ark?
- 10) Take two apples from three apples. What do you have?
- 11) If you drove a bus with 42 people on it from London and stopped at Watford to pick up 7 more, dropped off 5 passengers and at Luton dropped off 6 and picked up 4 and arrived at Edinburgh 20 hours later. What was the drivers name?
- 12) If a council worker has one hundred new houses from number one to one hundred, how many number nines will he need?

CP/M ON THE INTERAK COMPUTER

PART 2

by R.ELDRIDGE

This is part two of the article that tries to help you install an 8" single sided CP/M system onto the INTERAK 1 computer. Part 1 is in issue 3+4 JUNE/SEPT 1983. I will assume that you have acted upon the instructions given in part 1.

You have created a working copy of the standard distribution diskette. You have created a tape with two programs called GETSYS/PUTSYS.

To continue.....

Load the GETSYS/PUTSYS tape into the machine.

Load the WORK DISK onto drive A.

Use Zymon to E 8000.

This will load the standard distribution CP/M into the computers memory. as :-

```

Addr  Contents
3380  LOADER
3400  CCP
3C00  BDOS
4A00  BIOS
4D7F  end

```

The next step will require care and accuracy. Consider that we now have a copy of CP/M in store, it would run except that the LOADER and BIOS are for a different computer. What we are about to do is to overlay an INTERAK LOADER and BIOS to replace the ones recived on the standard distribution diskette. If you mistype even one byte you will quite probably have to completely restart the construction again. So please go slowly and carefully.

Using ZYMON very carefully enter the following code.

;20k INTERAK CBIOS AND LOADER

ADDR	DATA	LABEL	ASSEMBLER	COMMENTS
3380	21D300	LOADER:	LD HL,MESS	;Point to sign on
3383	CD21E0		CALL BOOT+21H	;Print sign on
3386	0E2E	LOAD:	LD C, '.'	;Get load start mark
3388	CD12E0		CALL BOOT+12H	;Print load start mark
338B	3E0D		LD A,RESTOR	;To seek track 0
338D	D380		OUT (DCOMM),A	;Command restore
338F	DB84	WZERO:	IN A,(DPOLL)	;Poll for DRQ INTRQ
3391	E601		AND 1	;Mask for INTRQ
3393	28FA		JR Z,WZERO	;Loop till restore done
3395	3E02		LD A,2	;Sector 2 on track 1
3397	D382	STARTS:	OUT (DSECT),A	
3399	324600		LD (SECTOR),A	
339C	3E9C		LD A,READMS	;Fetch read sector command
339E	324700		LD (COMMND),A	;Save read sector command
33A1	0EB3		LD C,DDATA	
33A3	1681		LD D,81H	


```

33A5 D380          OUT (DCOMM),A
33A7 DB84          DRQ:  IN A,(DPOLL)
33A9 A2            AND D
33AA 28FB          JR Z,DRQ
33AC EDA2          INI
33AE A2            AND D
33AF FAA700        JP M,DRQ
33B2 2B            DEC HL          ;Adjust address for INTRQ
33B3 DB81          IN A,(DTRAC)    ;Get track just read
33B5 324500        LD (TRACK),A    ;Save current track
33B8 3D            DEC A           ;Was it the last? ie track 1
33B9 C2C500        JP NZ,SEEK      ;If no seek track 1
33BC 21E200        LD HL,MESS1
33BF CD21E0        CALL BOOT+21H
33C2 C3004A        JP BIOS         ;Enter loader code
33C5 3E59          SEEK:  LD A,STEPIN ;Prepare to seek track 1
33C7 D380          OUT (DCOMM),A    ;Command seek track 1
33C9 DB84          WAIT1: IN A,(DPOLL) ;Poll for DRQ INTRQ
33CB E601          AND 1           ;Mask for INTRQ
33CD 28FA          JR Z,WAIT1      ;Loop till seek complete
33CF 3E01          LD A,1          ;First sector is 1 on track 1
33D1 18C4          JR STARTS       ;Branch to read track 1
33D3 00           MESS:  DEFB CR    ;Sign on
33D4 4C6F6164      DEFM 'Loader 20SD.2'
        65722032
        3053442E
        32
33E1 00           DEFB 0
33E2 0D0A00        MESS1: DEFB CR,LF,0

```

** THE ADDRESS CHANGES HERE TO 4A00H **

```

;Bios jump table
4A00 C39C4A        JP BOOT         ;Initialise post Cold start
4A03 C3B04A        WBOOTE: JP WBOOT ;Reload cp/m dos
4A06 C30CE0        JP BOOTR+0CH    ;Link to const
4A09 C30FE0        JP BOOTR+0FH    ;Link to conin
4A0C C312E0        JP BOOTR+12H    ;Link to conout
4A0F C3464B        JP LIST         ;Reg C to hardcopy device
4A12 C31BE0        JP BOOTR+1BH    ;Link to punch
4A15 C31EE0        JP BOOTR+1EH    ;Link to reader
4A18 C34A4B        JP HOME         ;Seek track 0
4A1B C3B94B        JP SELDSK       ;Select drive A,B,C or D
4A1E C34C4B        JP SETTRK       ;Staticise track 0-76
4A21 C3A44B        JP SETSEC       ;Staticise sector 1-26
4A24 C3AF4B        JP SETDMA       ;Staticise data source/dest
4A27 C3BE4B        JP READ         ;Read from disk
4A2A C3B54B        JP WRITE        ;Write to disk
4A2D C318E0        JP BOOTR+18H    ;Link to listst
4A30 C3A94B        JP STRAN        ;Translate sector skew = 6

```

;Drive parameter blocks.

```

4A33 824A        DPBASE: DEFW TRANS ;Translate table
4A35 0000        DEFW 0             ;Workspace A1
4A37 0000        DEFW 0             ;Workspace A2
4A39 0000        DEFW 0             ;Workspace A3

```

```

4A3B 314C      DEFW DIRBF      ;Directory buffer
4A3D 734A      DEFW DPBLK      ;Parameter block
4A3F 2D4D      DEFW CHK00      ;Drive A directory check
4A41 B14C      DEFW ALL00      ;Drive A allocation vectors

4A43 824A      DEFW TRANS      ;Translate table
4A45 0000      DEFW 0          ;Workspace B1
4A47 0000      DEFW 0          ;Workspace B2
4A49 0000      DEFW 0          ;Workspace B3
4A4B 314C      DEFW DIRBF      ;Directory buffer
4A4D 734A      DEFW DPBLK      ;Parameter block
4A4F 3D4D      DEFW CHK01      ;Drive B directory check
4A51 D04C      DEFW ALL01      ;Drive B allocation vectors

```

```

4A53 824A      DEFW TRANS      ;Translate table
4A55 0000      DEFW 0          ;Workspace C1
4A57 0000      DEFW 0          ;Workspace C2
4A59 0000      DEFW 0          ;Workspace C3
4A5B 314C      DEFW DIRBF      ;Directory buffer
4A5D 734A      DEFW DPBLK      ;Parameter block
4A5F 4D4D      DEFW CHK02      ;Drive C directory check
4A61 EF4C      DEFW ALL02      ;Drive C allocation vectors

```

```

4A63 824A      DEFW TRANS      ;Translate table
4A65 0000      DEFW 0          ;Workspace D1
4A67 0000      DEFW 0          ;Workspace D2
4A69 0000      DEFW 0          ;Workspace D3
4A6B 314C      DEFW DIRBF      ;Directory buffer
4A6D 734A      DEFW DPBLK      ;Parameter block
4A6F 5D4D      DEFW CHK03      ;Drive D directory check
4A71 0E4D      DEFW ALL03      ;Drive D allocation vectors

```

```

;Single density 8" disk definition block
;Disk total = 26 * 128 * 77 = 256256 Dtot
;Disk opsys = 26 * 128 * 2 = 6656 Dsys
;Disk data = 256256 - 6656 = 249600 Ddat
;Block size = 1024 BLS
;Disk size modulus = INT((Ddat / BLS) - 1) = 242 DSM
;Real data store = (DSM+1) * BLS = 248832 Rdat
;Allocation vector = (DSM/8)+1 = (242/8)+1 = 31.25 ALV

```

```

4A73 1A00      DPBLK: DEFW 26      ;SPT CP/M logical sectors
4A75 03         DEFB 3          ;BSH allocation block shift factor
4A76 07         DEFB 7          ;BLM block mask
4A77 00         DEFB 0          ;EXM extent mask (extent = 16k)
4A78 F200      DEFW 242        ;DSM disk size modulus
4A7A 3F00      DEFW 63         ;DRM number of directory entries-1
4A7C C0         DEFB 192        ;AL0 = ((DRM+1)*32)/BLS = 2
4A7D 00         DEFB 0          ;AL1 AL0AL1 = reserved blocks
4A7E 1000      DEFW 16         ;CKS = (DRM+1)/4 = 16 CSV
4A80 0200      DEFW 2          ;OFF reserved system tracks

```



```

;Sector translation table
4A82 01070D13 TRANS: DEFB 1,7,13,19 ;Translate logical to physical.
4A86 19050B11 DEFB 25,5,11,17 ;Skew = 6
4A8A 1703090F DEFB 23,3,9,15
4A8E 1502080E DEFB 21,2,8,14
4A92 141A060C DEFB 20,26,6,12
4A96 1218040A DEFB 18,24,4,10
4A9A 1016 DEFB 16,22

;Cold start initialisation.
4A9C 318000 BOOT: LD SP,80H ;Stack
4A9F 210E4C LD HL,SIGNON
4AA2 CD21E0 CALL BOOTR+21H ;Print 1st part
4AA5 320400 LD (CDISK),A ;Log drive A with BDOS
4AA8 324400 LD (DISKNO),A ;Stat drive A
4AAB 320300 LD (IOBYTE),A ;Null iobyte
4AAE 1867 JR GOCPM ;Initialise for cp/m

;Re-load cp/m at ctrl C or via 0000H
4AB0 318000 WBOOT: LD SP,80H ;Stack for warm load
;Save variables
4AB3 21334A LD HL,DPBASE
4AB6 11004E LD DE,BIOS+400H
4AB9 014000 LD BC,DPBLK-DPBASE
4ABC EDB0 LDIR
4ABE 212F4C LD HL,BEGDAT
4AC1 013E01 LD BC,ENDDAT-BEGDAT
4AC4 EDB0 LDIR
;Reload system
4AC6 3E0C LD A,0CH
4AC8 D385 OUT (DCONW),A ;Select drive A
4ACA 210034 LD HL,CCP ;Cp/m destination
4ACD 0E84 LD C,DPOLL ;Poll reg for DRQ INTRQ
;Initialised. Restore to track 0
4ACF 3E0D LD A,RESTOR ;To seek track 0
4AD1 D380 OUT (DCOMM),A ;Command restore
4AD3 ED78 WZERO: IN A,(C) ;Poll for DRQ INTRQ
4AD5 E601 AND 1 ;Mask for INTRQ
4AD7 28FA JR Z,WZERO ;Loop till restore done
;Read the full track
4AD9 3E02 LD A,2 ;Track 0 sectors 2 through 26
4ADB D382 OUT (DSECT),A
4ADD 1804 JR RDTRK0
4ADF 3E01 RDTRK1: LD A,1 ;Track 1 sectors 1 through 26
4AE1 D382 OUT (DSECT),A ;Give sector to FDCI
4AE3 3E9C RDTRK0: LD A,READMS
4AE5 D380 OUT (DCOMM),A
4AE7 ED50 WDRQ: IN D,(C)
4AE9 28FC JR Z,WDRQ
4AEB DB83 IN A,(DDATA)
4AED 77 LD (HL),A
4AEE 23 INC HL
4AEF FAE74A JP M,WDRQ
4AF2 2B DEC HL ;Adjust address for INTRQ
;Check for all tracks in

```

```

4AF3 DB81          IN A,(DTRAC)      ;Get track just read
4AF5 3D            DEC A              ;Was it the last? ie track 1
4AF6 2B0C          JR Z,WEND          ;If yes execute warm end
                                   ;Advance head to track 1
4AF8 3E59          LD A,STEPIN        ;Prepare to seek track 1
4AFA D380          OUT (DCOMM),A      ;Command seek track 1
4AFC ED78          WAIT1: IN A,(C)    ;Poll for DRQ INTRQ
4AFE E601          AND 1              ;Mask for INTRQ
4B00 2BFA          JR Z,WAIT1         ;Loop till seek complete
                                   ;Next get track 1 data
4B02 18DB          JR RDTRK1          ;Branch to read track 1
                                   ;Restore variables
4B04 21004E        WEND: LD HL,BIOS+400H
4B07 11334A        LD DE,DPBASE
4B0A 014000        LD BC,DPBLK-DPBASE
4B0D EDB0          LDIR
4B0F 112F4C        LD DE,BEGDAT
4B12 013E01        LD BC,ENDDAT-BEGDAT
4B15 EDB0          LDIR

                                   ;Cold and warm boot finals before cp/m begins
4B17 018000        GDCPM: LD BC,80H   ;For buffer initialisation
4B1A CD4F4B        CALL SETDMA        ;Set cp/m buffer address
4B1D DB40          IN A,(40H)         ;Discard any LK1 keys
4B1F 3EC3          LD A,0C3H         ;Ready JP opcode
4B21 320000        LD (0),A          ;Set 0000H = JP
4B24 21034A        LD HL,WBOOTE       ;Point to WBOOTE
4B27 220100        LD (1),HL         ;Form JP WBOOTE at 0000H
4B2A 320500        LD (5),A          ;Set 0005H = JP
4B2D 21063C        LD HL,BDOS         ;Point to BDOS
4B30 220600        LD (6),HL         ;Form JP BDOS at 0005H
4B33 323800        LD (38H),A        ;Set 0038H = JP
4B36 2106E0        LD HL,BOOTR+6     ;;Point to BOOT @ TRAPENT
4B39 223900        LD (39H),HL       ;Form JP TRAPENT at 0038H
4B3C 3A0400        LD A,(CDISK)       ;Get last logged disk
4B3F 324400        LD (DISKNO),A     ;Stat disk again
4B42 4F            LD C,A            ;Ready to give it to cp/m
4B43 C30034        JP CCP             ;Jump to CCP

                                   ;Hardcopy output link to boot rom
4B46 79            LIST: LD A,C       ;A has character to print
4B47 C315E0        JP BOOTR+15H      ;Give Acc to hardcopy

                                   ;Restore to track 0 on selected disk
4B4A 0E00          HOME: LD C,0       ;To set track = 0

                                   ;Seek to desired track.. C reg = track
4B4C 79            SETTRK: LD A,C     ;Track to seek to Acc
4B4D 324500        LD (TRACK),A       ;Staticise seek track.
4B50 060F          LD B,RETRY         ;Fetch retry
4B52 3EC0          RDID: LD A,READAD  ;Ready to read track ID
4B54 CD024C        CALL DISCOM        ;Read track ID
4B57 DB82          IN A,(DSECT)       ;Get track read
4B59 D381          OUT (DTRAC),A      ;Pass track to FDCI
4B5B DB80          IN A,(DSTAT)       ;Get Read ID status

```

```

4B5D E698          AND 98H          ;NRDY+RNF+CRC ERROR
4B5F 2018          JR NZ,IDERR       ;Jump if bad read
4B61 060F          LD B,RETRY        ;Track known reload retry
4B63 DB81          IN A,(DTRAC)      ;Get current track
4B65 B9            CP C              ;Compare desired track
4B66 C8            RET Z             ;Exit if already at track.
4B67 79            SEEKT: LD A,C      ;Seek track to Acc
4B68 D383          OUT (DDATA),A     ;Seek track to FDCI
4B6A 3E1D          LD A,SEEK        ;Ready to seek
4B6C CD024C        CALL DISCOM       ;Command a seek
4B6F DB80          IN A,(DSTAT)     ;Get seek status
4B71 E698          AND 98H          ;NRDY+SEEK ERROR+CRC ERROR
4B73 324800        LD (RXSTAT),A    ;Save status for errors
4B76 200C          JR NZ,SEEKER     ;Jump if bad seek
4B78 C9            RET              ;Exit seek complete
4B79 10D7          IDERR: DJNZ RDID  ;Retry-1, if retry<>0 re-read ID
4B7B 060F          LD B,RETRY        ;Cannot get ID so try restore
4B7D 3E0D          SEEK0: LD A,RESTOR ;Ready to restore
4B7F CD024C        CALL DISCOM       ;Command a restore
4B82 18E3          JR SEEKT         ;Retry seek from zero start
4B84 10F7          SEEKER: DJNZ SEEK0 ;Retry-1, if retry<>0 reseek
4B86 C336E0        JP DISKER        ;Perm seek error. Boot terminate.

```

```

;Select a drive. A,B,C or D
4B89 210000        SELDSK: LD HL,0   ;Return HL = 0 if error
4B8C 79            LD A,C            ;Get disk number A-D
4B8D 324400        LD (DISKNO),A    ;Staticise desired disk
4B90 FE04          CP 4              ;Check if A thru D
4B92 D0            RET NC           ;If no then return error
4B93 F60C          OR EIGHT+SD      ;Format for 8 SD
4B95 D385          OUT (DCONW),A    ;Select the drive
4B97 2600          LD H,0
4B99 79            LD A,C            ;Drive to Acc
4B9A 6F            LD L,A            ;HL = drive number 0-3
4B9B 29            ADD HL,HL         ;HL = drive * 2
4B9C 29            ADD HL,HL         ;HL = drive * 4
4B9D 29            ADD HL,HL         ;HL = drive * 8
4B9E 29            ADD HL,HL         ;HL = drive * 16
4B9F 11334A        LD DE,DPBASE     ;DE = drive parameter block
4BA2 19            ADD HL,DE         ;Index into parameter block
4BA3 C9            RET              ;Return HL = drive DPB

```

```

;Staticise desired sector
4BA4 79            SETSEC: LD A,C    ;Get sector to read/write
4BA5 324600        LD (SECTOR),A    ;Staticise sector
4BA8 C9            RET              ;Exit sector set up

```

```

;Translate logical sector to skewed by 6 sector
4BA9 EB            STRAN: EX DE,HL  ;Get translation base
4BAA 09            ADD HL,BC        ;Index into table
4BAB 6E            LD L,(HL)       ;Get physical sector
4BAC 2600          LD H,0          ;HL = physical sector
4BAE C9            RET              ;Exit sector translated

```

```

;Staticise desired memory address for read/write
4BAF 69      SETDMA: LD L,C      ;Get low order address
4BB0 60      LD H,B      ;Get high order address
4BB1 222F4C   LD (DMAAD),HL ;Staticise data address
4BB4 C9      RET      ;Exit data addr set up

;Write one sector to disk
4BB5 3EA3     WRITE: LD A,0A3H ;Z80 OUTI op-code
4BB7 32EB4B   LD (DAT+1),A ;Force data write
4BBA 3EAC     LD A,WRITS ;Set write data command
4BBC 1807     JR DCOM ;Execute write data

;Read one sector to store
4BBE 3EA2     READ: LD A,0A2H ;Z80 INI op-code
4BC0 32EB4B   LD (DAT+1),A ;Force data read
4BC3 3E8C     LD A,READS ;Set read data command

;Execute disk function, (read/write)
4BC5 324700   DCOM: LD (COMMND),A ;Save command
4BC8 3A4400   LD A,(DISKNO) ;Get current drive
4BCB F60C     OR EIGHT+SD ;Configure for BSD
4BCD D385     OUT (DCONW),A ;Select the drive on the FDCI
4BCF 060F     LD B,RETRY ;Load read/write retry count
4BD1 3A4600   COMGO: LD A,(SECTOR) ;Get sector to read/write
4BD4 D382     OUT (DSECT),A ;Pass this to the FDCI chip
4BD6 2A2F4C   LD HL,(DMAAD) ;Get data source/destination
4BD9 0E83     LD C,DDATA ;Disk data register port
4BDB 1681     LD D,81H ;Mask to flag DRQ INTRQ
4BDD 3A4700   LD A,(COMMND) ;Get command read/write
4BE0 D380     OUT (DCOMM),A ;Command a read/write
4BE2 C5       PUSH BC ;Stack retries
4BE3 0604     LD B,4 ;Load delay time
4BE5 10FE     WCOMM: DJNZ WCOMM ;Delay for command start
4BE7 C1       POP BC ;Recover retries
4BE8 1802     JR DRQ ;Enter read/write handler
4BEA EDA2     DAT: DEFB 0EDH,0A2H ;Read or write specifier
4BEC DB84     DRQ: IN A,(DPOLL) ;Poll for DRQ INTRQ
4BEE A2       AND D ;Set flags on poll data
4BEF FAEA4B   JP M,DAT ;If DRQ get/give data
4BF2 CAEC4B   JP Z,DRQ ;Loop till interrupt out
4BF5 DB80     IN A,(DSTAT) ;Get FDCI returned status
4BF7 E6DD     AND 0DDH ;Mask for errors
4BF9 C8       RET Z ;No errors so exit

;Read/write error detected
4BFA 10D5     DJNZ COMGO ;Else if retries, retry
;All command retries failed...Report error to user
4BFC 324800   LD (RXSTAT),A ;Terminal error, save status
4BFF C336E0   JP DISKER ;Enter boot error handler
4C02 324700   DISCOM: LD (COMMND),A ;Save command code
4C05 D380     OUT (DCOMM),A ;Give command in Acc to FDCI
4C07 DB84     QINTRQ: IN A,(DPOLL) ;Poll for DRQ INTRQ
4C09 E601     AND 1 ;Mask for INTRQ
4C0B 28FA     JR Z,QINTRQ ;Loop till interrupt out
4C0D C9       RET ;Exit command done

```

```

4C0E 0D0A      SIGNON: DEFB CR,LF      ;Sign on message
4C10 32306B20  DEFM '20k CP/M v2.2. IBIOS 20SD-6.'
      43502F4D
      2076322E
      322E2049
      42494F53
      20323053
      442D362E
4C2C 0D0A00    DEFB CR,LF,0
4C2F 00        DATASP: DEFB 0

```

You have finished typing in the code. Now use ZYMON to E 8003. This will write the customized code back to WORK DISK 1.

The following is a list of constants and is included for information only. To continue with the install just pass over this section.

```

;Ascii and system consants
= E000      BOOT      EQU 0E000H      ;Interak boot rom
= E036      DISKER    EQU BOOTR+36H   ;Disk error analiser
= 000D      CR        EQU 0DH         ;Ascii carriage return
= 000A      LF        EQU 0AH         ;Ascii line feed
= 0080      STACK     EQU 80H         ;Loader stack
= 000F      RETRY     EQU 15          ;Error retries
= 0008      EIGHT     EQU 8           ;8" disks
= 0004      SD        EQU 4           ;Single density

;Cpm/bdos/bios/factors
= 0014      MSIZE     EQU 20           ;Memory size in k's
= 0000      BIAS      EQU (MSIZE-20)*1024 ;Displacement factor
= 3400      CCP       EQU 3400H+BIAS   ;Start of CCP
= 3C06      BDOS      EQU CCP+806H     ;Start of BDOS
= 4A00      BIOS      EQU CCP+1600H    ;Start of BIOS
= 0004      CDISK     EQU 0004H       ;Current logged disk
= 0400      BLKSIZ    EQU 1024        ;BLS block size allocation

;Interak Scratchpad
= 0040      CURSOR    EQU 40H         ;Cursor if used
= 0042      KEYCDE    EQU CURSOR+2    ;Current key
= 0043      CURCAR    EQU KEYCDE+1    ;Old key
= 0044      DISKNO    EQU CURCAR+1    ;Current disk
= 0045      TRACK     EQU DISKNO+1    ;Track to seek
= 0046      SECTOR    EQU TRACK+1     ;Sector to seek
= 0047      COMMND    EQU SECTOR+1    ;Read/write/seek
= 0048      RXSTAT    EQU COMMND+1    ;Command status

;Disk controller ports
= 0080      DSTAT     EQU 80H         ;Disk status (read)
= 0080      DCOMM     EQU DSTAT       ;Disk command (write)
= 0081      DTRAC     EQU DCOMM+1     ;Disk track (read/write)
= 0082      DSECT     EQU DTRAC+1     ;Disk sector (read/write)
= 0083      DDATA     EQU DSECT+1     ;Disk data (read/write)
= 0084      DPOLL     EQU DDATA+1     ;Disk polling (read)
= 0085      DCONW     EQU DPOLL+1     ;Disk config out (write)
= 0085      DCONR     EQU DCONW       ;Disk user config (read)
= 0086      DOPTS     EQU DCONW+1     ;Disk options (read)

```

```

;Disk commands
= 00D0      RESET    EQU 0D0H      ;FDCI reset
= 000D      RESTOR   EQU 00DH      ;Seek track 0
= 001D      SEEK     EQU 01DH      ;Seek
= 008C      READS    EQU 08CH      ;Read sector
= 009C      READMS   EQU 09CH      ;Read multiple sectors
= 00AC      WRITS    EQU 0ACH      ;Write sector
= 00C0      READAD   EQU 0C0H      ;Read track ID
= 0059      STEPIN   EQU 59H       ;Step in one track
;Cbios data space
= 4C2F      BEGDAT   EQU DATASP
= 4C2F      DMAAD    EQU BEGDAT    ;Last dma address
= 4C31      DIRBF    EQU DMAAD+2    ;Directory check info
= 4CB1      ALL00    EQU DIRBUF+128 ;Drive A allocation vectors
= 4CD0      ALL01    EQU ALL00+31   ;Drive B allocation vectors
= 4CEF      ALL02    EQU ALL01+31   ;Drive C allocation vectors
= 4D0E      ALL03    EQU ALL02+31   ;Drive D allocation vectors
= 4D2D      CHK00    EQU ALL03+31   ;Drive A check vector
= 4D3D      CHK01    EQU CHK00+16   ;Drive B check vector
= 4D4D      CHK02    EQU CHK01+16   ;Drive C check vector
= 4D5D      CHK03    EQU CHK02+16   ;Drive D check vector
= 4D6D      ENDDAT   EQU CHK03+16
= 013E      DATSIZE  EQU ENDDAT-BEGDAT

```

INSTALLATION CONTINUES FROM HERE.

The customization is now complete but one more thing remains to be done before you can use your new system, you must replace ZYMON with a BOOT ROM. Remember the sequence of events that take place during a boot up of CP/M :-

Upon switch on the Z80 will be forced to execute code from E000H. At E000H it should find a BOOTstrap loader which will initialise the computer and then read in to memory location 0080H the first sector of track 1. When this is complete the Z80 will branch to 0080H and execute the code read in from sector 1 track 1 (LOADER). The LOADER will read from tracks 1 and 2 the complete CP/M operating system and place it at its designated store address. After that the Z80 will branch to the first location in it's customized BIOS and the CP/M system will sign on and be ready for use.

We have created the customized BIOS and LOADER but we still need to prepare the automatic bootstrap loader program.

THE BOOT ROM

Normally a CP/M system has its customized BIOS which allows CP/M to access the drives and peripherals of an alien system. Each system designer would produce a customized BIOS once only and this would be used on all of that type of computer. The INTERAK computer is different and cannot simply use this method. The reason for this difference is the greater flexibility that the INTERAK offers to it's user. Consider that you have constructed a LOADER BIOS pair to drive the VDU-2K, a teletype and a parallel keyboard. You have been using your system for about a year and have accumulated 50 or 60 diskettes. If now Greenbank produced an 80 by 26 serial terminal and you choose to change over to it. You

would first write a new BIOS and then construct a complete new CP/M, in much the same way as we have just done, then each time one of your previous diskettes was required you would transfer that system to its system tracks. Very messy and upsetting, especially if you are hand building your customized components of the new system. Also, all systems require a boot rom, this usually consists of enough code to read in one sector from track zero and is probably of the order of 200 bytes or less. The remaining ROM space is unused.

My concept of the INTERAK CP/M is to make use of that EPROM to retain the flexibility of the computer and yet make it unnecessary to rewrite the disk operating system customized code when improvements or upgrades are made. This is achieved very simply by writing the LOADER BIOS pair such that all peripheral transfers are passed to the BOOT ROM for final processing. In this way the device dependant drivers are held by the BOOT ROM and to upgrade your system requires only that the BOOT ROM be altered or replaced.

Greenbank currently can supply three customized BOOT ROMS which will suffice for the above system. If you prefer to burn your own I will print the code for BOOT ROM 0301-0100-018401, the recommended INTERAK 1 version, in the next issue.

Here is a brief description of the currently available BOOT ROMS :-

BOOT ROM IDENTIFIER NUMBER = 0301-0100-018401

This is recommended for 8" drives on the INTERAK computer

VDU-2K,

MEMORY MAPPED 64*24 @F000H

LK1 KEYBOARD,

PRINTER,

DATA OUT PORT 7, STATUS IN PORT 6 (As for ZYBASIC)

BOOT ROM IDENTIFIER NUMBER = 0502-0200-018405

TERMINAL VDU, DATA OUT PORT 1, STATUS IN PORT 0, TBMT IS BIT 7

TERMINAL KEYBOARD, DATA IN PORT 1, STATUS IN PORT 0, DAV IS BIT 6

PRINTER, DATA TO PORT 3, STATUS IN PORT 2, TBMT IS BIT 7

BOOT ROM IDENTIFIER NUMBER = 0601-0100-048405

INTELGRAPH VDU, DATA OUT PORT 1, STATUS IN PORT 0, TBMT IS BIT 7

LK1 KEYBOARD

PRINTER,

DATA OUT PORT 7, STATUS IN PORT 6, TBMT IS BIT 7

As a bonus a side effect of the above policy is that anybody's diskettes will boot up on any INTERAK, even if each has different BOOT ROMS. I have found this quite useful when working with other people to achieve some common aim. So, insert the appropriate BOOT ROM in place of ZYMON, switch on the computer, insert your newly created diskette and CP/M should sign on.

In the next issue I will print the current version of the standard INTERAK BOOT ROM. (VDU-2K, LK1, PRINTER 1).

THE END

COMPETITION NOTES

In order to encourage users to send in to the Interaktion User Group examples of software they have written, developed, or implemented for the Interak Computer, a small prize has been offered by Greenbank Electronics. (The work need not be original or unpublished, but don't send in anything which will result in your going to gaol!)

Up to one prize a month will be offered, until further notice, and each will be an Interak Bare Board. In the absence of entries of sufficient merit (as judged by Greenbank Electronics) the prize will not be awarded, and exceptionally more than one prize per month will be given.

Conditions:

1. The prizewinners must be published in the Interaktion Newsletter.
2. The software must be made available for distribution to other users, at a price to be agreed, or preferably simply for the cost of distribution.
3. The prize can be exchanged for another board by an Interak supplier, at the supplier's option; it cannot be exchanged for goods or credit. (It can of course be sold or given away.)

David Parkins

INTERAKTION BOOK LIBRARY

This section is to give members access to a wide range of books on computing and electronics. The only cost to the member is that of postage. Books may be borrowed for up to 3 weeks, and are available from the User Group address. Member Dick Bowyer is acting as librarian for now. At present the books available are:

LANGUAGE BOOKS

TRS 80 Assembly Language	
Programming	Radio Shack
Z80 Assembly Language Programming	
Manual	Zilog
A Course in Basic Programming	Sinclair
Making the Most of your ZX 80	Tim Hartnell
30 Hour Basic	C.Prigmore
Basic for Home Computers a	
Self-Teaching Guide	B.Albrecht, L.Finkel & J.Brown
Course in Standard Coral 66	J.D.Halliwel & T.A Edwards
Simple Pascal	J.McGregor & A.Watt
Lecture Notes in Computer Science	
Pascal User Manual and Report	K.Jensen & N.Wirth

DATA BOOKS

Mostek 1982/1983 Microelectronic
Data Book (memory/CPU/Peripherals) Mostek
Memory Data Book and Designers Guide
1980 Mostek
Bytewyde Memory Data Book 1981 Mostek
National Semiconductor Memory
Data Book 1980 National
National Semiconductor Interface
Data Book 1980 National
TTL Data Book National
The European Selection
(memory/interface/linear) Motorola

GENERAL & ELECTRONICS

Computer Technology for Technicians
and Technical Engineers Vol. 1 R. Watkin
Electronic Computers Made Simple H. Jacobowitz
Test Instruments for Electronics
(how to build test instruments) M. Clifford
Practical Test Instruments
You Can Build W. Green
How to Troubleshoot & Repair
Electronic Test Equipment M. Horowitz
Computers and the Social Sciences A. Brier & I. Robinson

MANUALS etc.

Epson MX-80 Type II Operation Manual .. Epson
Newbury 8000 Series VDU Terminal
1,yes,12,match,1hr,9,70,white,0,2,yours,20
Operator Instruction Manual Newbury Labs
Electronics Projects Index
(A descriptive guide to 2500 projects
published in popular magazines.
Quite old now.) Polytechnic
Why Do You Need a Personal Computer? .. Leventhal & Straffars
Computer Programming in the Classroom . B.J.Jackson
TABS Accounting Business Systems
User Guide Vol 1 TABS
Easy Add-on Projects for Spectrum,
ZX-81, Jupiter Ace Owen Bishop
6502 Games Rodney Zaks

All books have been donated by users (a lot from Greenbank). If you
have any books etc. surplus to requirements please let me have them.

Richard Bowyer

SOFTWARE LIBRARY

(Please enquire for cost of postage, and items marked POA)

NAME	DESCRIPTION	AUTHOR	CODE	SUPP.	SCREEN	COST
----	-----	-----	----	----	-----	-----
ZYMON 2	INTERAK monitor	BE	MC	GB	A	GB
ZYBASIC 2	INTERAK BASIC	NK	MC	GB	A	GB
ZYMON 2	INTERAK monitor	BE	MC	UG	C	POA
ZYBASIC 2	INTERAK BASIC	NK	MC	UG	C	POA
XTAL BASIC	14K BASIC	XL	MC	UG	A	£40
XTAL BASIC	14K BASIC	XL	MC	UG	C	£40
FIGFORTH	Forth Compiler	CD	MC	UG	A	£15
FIGFORTH	Forth Compiler	CD	MC	UG	C	£15
ASM 32	Editor Assembler	NK	MC	UG	A	£10
ASM 64	Editor Assembler	NK	MC	UG	C	£10
HC DISASS	Simple Disassembler	HC	MC	UG	A	£ 3
REVAS	Better Disassembler	DP	MC	UG	A	POA
MEGABUG	Debug/Training Package	RD	MC	UG	C	£13
VELTEXT	Text Editor	PV	MC	UG	A	£ 5
VELTEXT	Text Editor	PV	MC	UG	C	£ 5
Lander	Lander Game	PV	XL	UG	C	\
Towers	Towers Puzzle	PV	XL	UG	C	PP
Crazy Maze	"3D" Maze Game	PV	XL	UG	C	/
Avalanche	Blob Dodging Game	DB	ZB2	UG	A	PP
Monster Mash	Maze Game	BE	ZB2	UG	A	PP
Graph	Graph Plotter	MC	ZB2	UG	A	PP
Rakovsky	Computer Chess (6 levs)	NK	MC	UG	A	\
AC10.XX	(Chess Character EPROM for VDU-K) -	NK	NA	UG	A	/
Rakovsky	Computer Chess (6 levs)	NK	MC	UG	C	\
AC10.XX	(Chess Character EPROM for VDU-2K) -	NK	NA	UG	C	/
Happy Sums	Fun maths	PV	ZB2	UG	A	PP
Hangman	Spelling game	PV	ZB2	UG	A	PP
0's and X's	Game	PV	ZB2	UG	A	PP
Pools Pick	Random Draw Selector	PV	ZB2	UG	A	PP
Count	Learn to count	PV	ZB2	UG	A	PP
Dice Pontoon	Simple Game	PV	ZB2	UG	A	PP

Key: MC machine code. Screen: A 32 x 24 VDU-K
 ZB2 ZYBASIC. B 64 x 16 VDU-1K
 XL XTAL BASIC. C 64 x 24 VDU-2K
 GB Greenbank.
 UG User Group.
 PP Postage & Packing.
 POA Please enquire (Price on Application).
 NK Unknown

Orders and enquiries to Interaktion User's Group
 Pete Vella,
 19 Ford Drive,
 Yarnfield,
 Nr Stone,
 STAFFS ST15 0RP.

GENERAL PURPOSE VDU 2K DRIVER.
BY R.ELDRIDGE.

```
;Conout,(sub),sends the C reg to the VDU.
;Treats the VDU-2k as a scrolling terminal.
;Put ASCII code to be printed in the C register and CALL CONOUT
;If C reg 0DH = CR then do carriage return.
;If C reg 0AH = LF then do line feed.
;If C reg 08H = BS then do backspace.
;If C reg 7FH = DELETE then do backspace
;Else pass C reg to cursor position, adjust cursor, return to caller.
;Originated at 0000. Alter addresses to reposition or reassemble
;under ASM64.
```

```
>0000                ORG 0

>0020                SPACE    EQU 20H
>000D                CR      EQU 0DH
>000A                LF      EQU 0AH
>0008                BS      EQU 08H
>007F                DEL     EQU 7FH

>F000                LINE1    EQU 0F000H
>0018                LINES    EQU 24
>0040                COLS     EQU 64
>F5C0                LINE24   EQU LINE1+(COLS*(LINES-1))

0000  C5              CONOUT:  PUSH BC          ;Save BC
0001  D5              PUSH DE          ;Save DE
0002  E5              PUSH HL          ;Save HL
0003  2A6200          LD HL,(CURSOR)    ;HL = cursor
0006  3A6400          LD A,(CURCAR)    ;Character under cursor
0009  77              LD (HL),A        ;Put cursor character back
000A  79              LD A,C          ;Get character to display
000B  CBBF            RES 7,A          ;Reset parity bit
000D  FE0D            CP CR            ;Is it carriage return?
000F  200E            JR NZ,NCR        ;Branch if not CR

0011  21C0F5          LD HL,LINE24     ;Point to bottom line
0014  7E              LD A,(HL)        ;Get this character in A
0015  326400          LD (CURCAR),A    ;Save A as curcar
0018  226200          EXIT:  LD (CURSOR),HL ;Save cursor position
001B  E1              POP HL           ;Restore HL
001C  D1              POP DE           ;Restore DE
001D  C1              POP BC           ;Restore BC
001E  C9              RET              ;Exit done

001F  FE0A            NCR:  CP LF        ;Is it line feed?
0021  2021            JR NZ,NLF        ;Branch if not LF

0023  1100F0          LFEED:  LD DE,LINE1 ;Point to screen top
0026  2140F0          LD HL,LINE1+COLS ;Point to line 2 start
0029  01C005          LD BC,(LINES-1)*COLS ;Number to move
002C  EDB0            LDIR             ;Scroll up
002E  21C0F5          LD HL,LINE24     ;Load bottom line
```

```

0031 0640          LD B,COLS          ;Load line length
0033 3E20          LD A,SPACE        ;Space code
0035 77           CL24: LD (HL),A     ;Blank first
0036 23           INC HL             ;Advance pointer
0037 10FC          DJNZ CL24         ;Blank bottom line
0039 21C0F5        LD HL,LINE24      ;Load bottom line
003C 7E           SAVCUR: LD A,(HL)   ;Get cursor character
003D 326400        LD (CURCAR),A     ;Save it in curcar
0040 365F          LD (HL), '-'      ;Print the cursor
0042 18D4          JR EXIT           ;Branch to exit

0044 FE08          NLF: CP BS         ;Is it backspace?
0046 2005          JR NZ,NBS         ;Branch if not
0048 3620          DELIT: LD (HL),SPACE ;Blot out character
004A 2B           DEC HL             ;Backspace pointer
004B 18EF          JR SAVCUR         ;Branch to exit
004D FE7F          NBS: CP DEL        ;Is it delete?
004F 28F7          JR Z,DELIT        ;Do as backspace if delete
0051 77           LD (HL),A         ;Print code in A
0052 23           INC HL             ;Advance the cursor
0053 1100F6        LD DE,LINE24+COLS ;Screen end
0056 CD5D00        CALL COMPAR        ;Does cursor = end?
0059 28C8          JR Z,LFEED        ;Branch if end
005B 18DF          JR SAVCUR         ;Branch if not end

005D B7           COMPAR: OR A        ;Clear carry
005E ED52          SBC HL,DE         ;Compare HL:DE set flags
0060 19           ADD HL,DE         ;Restore HL
0061 C9           RET               ;Return flags on compare

0062 C0F5          CURSOR: DEFW LINE24 ;Cursor address
0064 20           CURCAR: DEFB 20H   ;Character under cursor

```

END

The above code can be the basis of any program that needs to pass ASCII codes to the VDU 2K. Perhaps a new monitor program or a CP/M installation. Just adjust the addressing to move the code to a new position in the machine store.

THE END

LETTERS TO THE EDITOR

Please write with comments, ideas, complaints and suggestions. Name and address must be enclosed. Responsibility for views and comments expressed cannot be held by the editor as members letters are published with as few changes possible.

Dear Ed,

Computer people who live in the SOUTH EAST LONDON area may be interested in joining the DARTFORD COMPUTER CLUB.

This is a small group, of all ages and sexes, who enjoy computers and computer science. There is a meeting every week and throughout the year many social events take place.

The club includes all micro-based machines, and the meetings allow you to get hands-on experience with several different computers.

If any of your members are interested will they please contact me for details of the next meeting, or just to have a chat about the club.

GARY MEAD, 52 KNOCE ROAD, DARTFORD, KENT. 01-854-9081, daytime.

Dear Ed,

I regret that I shall be unable to invite you, gentlemen, to my cheese cube and thimble-full of wine party to celebrate the commissioning of my INTERAK 0.5.

I had a list of 35 guests, so I asked it how I should arrange the seating. (Page 2-11) - It just threw the question back at me - "HOW". Admittedly it is more polite than the MICON CNC machine computer, which says "WHAT?".

The best that ZYBASIC can do for me is, 868,330,000,000,000,000,000,000,000,000,000,000 for 33 people, so I had to cross two names off the list - guess whose?

Thank you for the computer, I shall now get back to working through the ZYBASIC manual and see what else I don't understand!!

DAVE BALLAM, 13 WOOLIFERS AVE, CORRINGHAM, ESSEX. SS17 9AU.

Dear Ed,

Cri_de_Coeur

I am receiving many pleas for help from people who have bought and constructed Interak 1 (but who are otherwise quite intelligent), and don't know what to do with it. They are a bit like budding artists who have bought a canvas, paints and an easel to paint themselves a masterpiece, but don't quite know where to begin.

Their main need seems to be for help writing machine code, or assembler code. They can do the simple stuff like loading a register with a number, and then seeing if the register contains the number they have loaded; they are aware they are not quite ready yet to write a 24K Pascal compiler or the like, but they want a job to do.

I try and help by asking the question, "When you first decided to build a computer what job did you think it was going to do?", and based on the reply I try and suggest an appropriate simple project to make a start. E.g. if a accounting use was proposed, the first simple software project could be to write a program to accept two numbers, add them up and put the result on the screen; if someone wanted eventually to build a robot, I would suggest his first software project would be to turn a light or relay on and off. (There are some handy lights and relays built into the computer which many people don't realise they can use for themselves, for instance an indicator lights which can be turned on and off by software are to be found on the DTI-1 card; and the two tape motor control relays can be used for other purposes, e.g. turning a model railway engine on, changing points etc. - provided the current is not too much for the miniature relays to carry that is - take care!)

But I'm no educationalist, and I know lots of users are. Can they help by suggesting an elementary, and I mean elementary - that's where the need is greatest, scheme of work for an absolute beginner to follow from the very first day he switches on his Interak?

DAVID.M.PARKINS, GREENBANK ELECTRONICS, 92 NEW CHESTER ROAD, NEW FERRY, WIRRAL, MERSEYSIDE, L62 5AG.

Dear ED,

Some of your members may like to quote Michael Faraday when dealing with unimpressed members of the family. When he was asked about the usefulness of his discoveries in electromagnetism he replied:-

"What use is a new born baby?"

NAME AND ADDRESS WITHHELD.